Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A system for providing a [[Java]] JAVA code release infrastructure with granular code patching, comprising:

one or more [[Java]] JAVA code patches, each comprising at least one resource unit, each resource unit comprising metadata and file components;

one or more [[Java]] JAVA code libraries, each comprising at least one such resource unit;

a patch tool, comprising:

a compare module that determines which units in the JAVA code libraries are outdated by comparing the metadata for each such resource unit in the [[Java]] JAVA code patches to the metadata for each such corresponding resource unit in the [[Java]] JAVA code libraries; and

a merge module merging each such resource unit in the [[Java]] JAVA code patches into the [[Java]] JAVA code libraries for each such corresponding resource unit that is out-of-date.

2. (currently amended) A system according to Claim 1, further comprising:

an extract module extracting at least one resource unit from the [[Java]] JAVA code libraries and modifying one or more [[Java]] JAVA archive files that are out-of-date with the at least one extracted resource unit.

3. (currently amended) A system according to Claim 1, further comprising:

a sign module signing the [[Java]] JAVA archive files using a digital certificate.

4. (currently amended) A system according to Claim 1, wherein the one or more [[Java]] JAVA archive files are modified through at least one of creation, revision or deletion.

5. (currently amended) A system according to Claim 1, further comprising:

a source repository storing the source file components;
a staged patch repository storing the one or more [[Java]] JAVA code patches; and
a staged code repository organizing the one or more [[Java]] JAVA code libraries and the [[Java]] JAVA archive files.

6. (currently amended) A system according to Claim 1, further comprising:
a resource unit generator processing the file components into at least one such resource unit; and
a packager packaging at least one such resource unit into one or more of the [[Java]] JAVA code patches.

7. (currently amended) A system according to Claim 6, further comprising:
stored [[Java]] JAVA source code provided as the file components.

8. (currently amended) A system according to Claim 7, further comprising:
a compiler compiling at least one [[Java]] JAVA source code file into one or more [[Java]] JAVA classes; and
a resource unit packager module storing the [[Java]] JAVA classes into at least one such resource unit as the file components.

9. (currently amended) A system according to Claim 6, further comprising:
at least one of ~~non-Java~~ non-JAVA source and derived code provided as the file components.

10. (original) A system according to Claim 6, further comprising:
third party code provided as the file components.

11. (original) A system according to Claim 6, further comprising:
a metadata generator generating the metadata for each such resource unit; and
a resource unit packager module storing the generated metadata into the resource unit.

12. (original) A system according to Claim 11, wherein the metadata comprises at least one of a unique identifier and a version attribute.

13. (currently amended) A system according to Claim 1, further comprising:
a compare module using a set of rules allowing one of an older resource unit to be replaced by a newer resource unit and a newer resource unit to be replaced by an older resource unit to back out a previously-applied [[Java]] JAVA code patch.

14. (currently amended) A system according to Claim 1, further comprising:
one or more [[Java]] JAVA archive files, each comprising at least one resource unit corresponding to one such resource unit in the [[Java]] JAVA code libraries; and
a patch tool referencing [[Java]] JAVA archive file definitions which each correspond to one or more of the [[Java]] JAVA archive files.

15. (currently amended) A system according to Claim 14, further comprising:
an extract module extracting the resource units from the [[Java]] JAVA code libraries into the [[Java]] JAVA archive files for each such corresponding resource unit that is out-of-date.

16. (currently amended) A system according to Claim 15, further comprising:
an extract module referencing third party [[Java]] JAVA code libraries not maintained as part of the infrastructure.

17. (currently amended) A system according to Claim 1, further comprising:
[[Java]] JAVA code libraries implemented as a portable virtual file system which can be used directly by a [[Java]] JAVA Virtual Machine.

18. (currently amended) A system according to Claim 1, further comprising:
a machine portable infrastructure providing support for [[Java]] JAVA language features by encapsulating [[Java]] JAVA inner classes, nested directory structures, native class names, and native character set.

19. (currently amended) A method for providing a [[Java]] JAVA code release infrastructure with granular code patching, comprising:

providing one or more [[Java]] JAVA code patches, each comprising at least one resource unit, each resource unit comprising metadata and file components;

patching one or more [[Java]] JAVA code libraries, each comprising at least one such resource unit;

comparing the metadata for each such resource unit in the [[Java]] JAVA code patches to the metadata for each such corresponding resource unit in the [[Java]] JAVA code libraries; and

merging each such resource unit in the [[Java]] JAVA code patches into the [[Java]] JAVA code libraries for each such corresponding resource unit that is out-of-date.

20. (currently amended) A method according to Claim 19, further comprising:

extracting at least one resource unit from the [[Java]] JAVA code libraries and modifying one or more [[Java]] JAVA archive files that are out-of-date with the at least one extracted resource unit.

21. (currently amended) A method according to Claim 19, further comprising:

signing the [[Java]] JAVA archive files using a digital certificate.

22. (currently amended) A method according to Claim 19, wherein the one or more [[Java]] JAVA archive files are modified through at least one of creating, updating or deleting.

23. (currently amended) A method according to Claim 19, further comprising:

storing source file components into a source repository;

storing one or more [[Java]] JAVA code patches into a staged patch repository; and

organizing one or more [[Java]] JAVA code libraries and the [[Java]] JAVA archive files into a staged code repository.

24. (currently amended) A method according to Claim 19, further comprising:

processing the file components into at least one such resource unit; and

packaging at least one such resource unit into one or more of the [[Java]] JAVA code patches.

25. (currently amended) A method according to Claim 24, further comprising:
providing [[Java]] JAVA source code as the file components.

26. (currently amended) A method according to Claim 25, further comprising:
compiling at least one [[Java]] JAVA source code file into one or more [[Java]] JAVA classes; and
storing the [[Java]] JAVA classes into at least one such resource unit as the file components.

27. (currently amended) A method according to Claim 24, further comprising:
providing at least one of ~~non-Java~~ non-JAVA source and derived code as the file components.

28. (original) A method according to Claim 24, further comprising:
providing third party code as the file components.

29. (original) A method according to Claim 24, further comprising:
generating the metadata for each such resource unit; and
storing the generated metadata into the resource unit.

30. (original) A method according to Claim 29, wherein the metadata comprises at least one of a unique identifier and a version attribute.

31. (currently amended) A method according to Claim 19, further comprising:
using a set of rules to allow one of an older resource unit to be replaced by a newer resource unit and a newer resource unit to be replaced by an older 3 resource unit to back out a previously-applied [[Java]] JAVA code patch.

32. (currently amended) A method according to Claim 19, further comprising:
providing one or more [[Java]] JAVA archive files, each comprising at least one resource unit corresponding to one such resource unit in the [[Java]] JAVA code libraries; and

referencing [[Java]] JAVA archive file definitions which each correspond to one or more of the [[Java]] JAVA archive files.

33. (currently amended) A method according to Claim 32, further comprising:
extracting the resource units from the [[Java]] JAVA code libraries for each such corresponding resource unit that is out-of-date.

34. (currently amended) A method according to Claim 33, further comprising:
referencing third party [[Java]] JAVA code libraries not maintained as part of the infrastructure.

35. (currently amended) A method according to Claim 19, further comprising:
implementing [[Java]] JAVA code libraries as a portable virtual file system which can be used directly by a [[Java]] JAVA Virtual Machine.

36. (currently amended) A method according to Claim 19, further comprising:
providing a machine portable infrastructure supporting [[Java]] JAVA language features by encapsulating [[Java]] JAVA inner classes, nested directory structures, native class names, and native character set.

37. (original) A computer-readable storage medium holding code for performing the method of Claim 19.

38. (currently amended) A system for patching staged code in a staged [[Java]] JAVA code release infrastructure, comprising:
a staged code repository maintaining one or more staged [[Java]] JAVA code libraries, each staged [[Java]] JAVA code library comprising at least one resource unit, each resource unit comprising metadata and file components;
a staged patch repository storing one or more [[Java]] JAVA code patches, each [[Java]] JAVA code patch comprising at least one resource unit corresponding to one such resource unit specified in a [[Java]] JAVA code patch definition; and

a patch tool accessing one or more [[Java]] JAVA code patches in the staged patch repository, comprising:

a compare module comparing the metadata for each resource unit in the [[Java]] JAVA code patches to the metadata in the staged [[Java]] JAVA code libraries for each such corresponding resource unit; and

a merge module merging each resource unit in the [[Java]] JAVA code patches into the staged [[Java]] JAVA code libraries for each such corresponding resource unit that is out-of-date.

39. (currently amended) A system according to Claim 38, further comprising:

an extract module referencing [[Java]] JAVA archive file definitions which each correspond to a staged [[Java]] JAVA archive file, each staged [[Java]] JAVA archive file comprising at least one resource unit corresponding to one such resource unit in the staged [[Java]] JAVA code libraries.

40. (currently amended) A system according to Claim 39, further comprising:

an extract module extracting one such resource unit from the staged [[Java]] JAVA code libraries into the staged [[Java]] JAVA archive files for each such corresponding resource unit that is out-of-date.

41. (currently amended) A system according to Claim 40, further comprising:

a sign module creating a digital signature for the staged [[Java]] JAVA archive files using a digital certificate.

42. (currently amended) A method for patching staged code in a [[Java]] JAVA code release infrastructure, comprising:

maintaining one or more staged [[Java]] JAVA code libraries in a staged code repository, each staged [[Java]] JAVA code library comprising at least one resource unit, each resource unit comprising metadata and file components;

accessing one or more [[Java]] JAVA code patches in a staged patch repository, each [[Java]] JAVA code patch comprising at least one resource unit corresponding to one such resource unit specified in a [[Java]] JAVA code patch definition;

comparing the metadata for each resource unit in the [[Java]] JAVA code patches to the metadata in the staged [[Java]] JAVA code libraries for each such corresponding resource unit; and

merging each resource unit in the [[Java]] JAVA code patches into the staged [[Java]] JAVA code libraries for each such corresponding resource unit that is out-of-date.

43. (currently amended) A method according to Claim 42, further comprising:

referencing [[Java]] JAVA archive file definitions which each correspond to a staged [[Java]] JAVA archive file, each staged [[Java]] JAVA archive file comprising at least one resource unit corresponding to one such resource unit in the staged [[Java]] JAVA code libraries.

44. (currently amended) A method according to Claim 43, further comprising:

extracting one such resource unit from the staged [[Java]] JAVA code libraries into the staged [[Java]] JAVA archive files for each such corresponding resource unit that is out-of-date.

45. (currently amended) m A method according to Claim 44, further comprising:

creating a digital signature for the staged [[Java]] JAVA archive files using a digital certificate.

46. (original) A computer-readable storage medium holding code for performing the method of Claim 42.

47. (currently amended) A system for generating [[Java]] JAVA code patches in a [[Java]] JAVA code release infrastructure, comprising:

a source code repository maintaining one or more source files; and

a patch generator generating one or more [[Java]] JAVA code patches, each comprising at least one resource unit, each resource unit comprising metadata and file components specified in [[Java]] JAVA code patch definitions.

48. (currently amended) A system according to Claim 47, further comprising:

a resource unit generator processing the file components into at least one such resource unit; and

a packager packaging at least one resource unit into one or more of the [[Java]] JAVA code patches.

49. (currently amended) A system according to Claim 48, further comprising:
one or more [[Java]] JAVA source code files provided as the file components.

50. (currently amended) A system according to Claim 49, further comprising:
a compiler compiling the [[Java]] JAVA source code into one or more [[Java]] JAVA classes; and
a resource unit packager module storing the [[Java]] JAVA classes into at least one such resource unit as file components.

51. (currently amended) A system according to Claim 48, further comprising:
at least one of ~~non-Java~~ non-JAVA source and derived code provided as the file components.

52. (original) A system according to Claim 48, further comprising:
staged third party code provided as the file components.

53. (original) A system according to Claim 48, further comprising:
a metadata generator generating the metadata for each such resource unit;
and
a resource unit packager module storing the generated metadata into one such resource unit.

54. (currently amended) A system according to Claim 53, wherein the metadata comprises at least one of a unique identifier and a version attribute.

55. (currently amended) A method for generating [[Java]] JAVA code patches in a [[Java]] JAVA code release infrastructure, comprising:
maintaining one or more source files in a source code repository; and

generating one or more [[Java]] JAVA code patches, each comprising at least one resource unit, each resource unit comprising metadata and file components specified in [[Java]] JAVA code patch definitions.

56. (currently amended) A method according to Claim 55, further comprising:
processing the file components into at least one such resource unit; and
packaging at least one resource unit into one or more of the [[Java]] JAVA code patches.

57. (currently amended) A method according to Claim 56, further comprising:
providing one or more [[Java]] JAVA source code files as the file components.

58. (currently amended) A method according to Claim 57, further comprising:
compiling the [[Java]] JAVA source code into one or more [[Java]] JAVA classes; and
storing the [[Java]] JAVA classes into at least one such resource unit as file components.

59. (currently amended) A method according to Claim 56, further comprising:
providing at least one of ~~non-Java~~ non-JAVA source and derived code as the file components.

60. (original) A method according to Claim 56, further comprising:
providing staged third party code as the file components.

61. (original) A method according to Claim 56, further comprising:
generating the metadata for each such resource unit; and
storing the generated metadata into one such resource unit.

62. (original) A method according to Claim 61, wherein the metadata comprises at least one of a unique identifier and a version attribute.

63. (original) A computer-readable storage medium holding code for performing the method of Claim 55.